

**GASIC 8– Seguridad en el Desarrollo y Adquisición de Software**

**Preguntas de Auditoría**

**Protección del Código y Software**

**Competencias Básicas del Auditor**

1. **Conocimiento Profundo en Desarrollo de Software Seguro:** Entender los principios de seguridad en el desarrollo de software, incluyendo buenas prácticas de codificación, diseño seguro, y consideraciones de seguridad en el ciclo de vida del desarrollo.
2. **Experiencia en Identificación de Vulnerabilidades:** Tener habilidades para identificar y evaluar vulnerabilidades en el código y el diseño del software, incluyendo análisis estático y dinámico, pruebas de penetración y revisión de arquitecturas.
3. **Conocimiento en Criptografía y Protección de Datos:** Comprender los principios de la criptografía y cómo se aplica en la protección de datos sensibles dentro del software, incluyendo la gestión segura de claves y la integridad de los datos.
4. **Habilidades en Pruebas de Seguridad y Evaluación de Riesgos:** Ser capaz de realizar pruebas de seguridad exhaustivas para evaluar la resistencia del software frente a posibles ataques, así como evaluar los riesgos asociados con el software.
5. **Conocimientos en Cumplimiento y Normativas:** Estar familiarizado con los requisitos de cumplimiento y normativas relevantes en el ámbito de la seguridad del software, como GDPR, PCI DSS, y normas ISO/IEC.
6. **Capacidad Analítica:** Ser capaz de analizar datos de seguridad y tendencias para identificar posibles amenazas y vulnerabilidades emergentes en el software.
7. **Habilidades de Comunicación Técnica:** Ser capaz de comunicar de manera efectiva los hallazgos de seguridad a los desarrolladores, gerentes y otros interesados, y proponer recomendaciones claras y prácticas.
8. **Conocimientos en Gestión de Incidentes y Respuesta a Incidentes:** Entender los procedimientos y prácticas para la gestión de incidentes de seguridad en el software, incluyendo la capacidad de respuesta rápida y eficaz.

# Visión General

En el ámbito de la ciberseguridad durante el desarrollo y adquisición de software implica la implementación de prácticas y medidas destinadas a salvaguardar la integridad, confidencialidad y disponibilidad de las aplicaciones. Este enfoque se extiende desde la etapa inicial de diseño hasta la implementación y mantenimiento continuo, abordando aspectos cruciales como el análisis estático y dinámico del código para identificar y mitigar vulnerabilidades, la aplicación de buenas prácticas de codificación y la integración de medidas de seguridad en cada fase del ciclo de vida del software. Además, implica la adopción de técnicas de prueba de seguridad, la gestión adecuada de identidades y accesos, la implementación de controles criptográficos y la evaluación de diseños desde una perspectiva de seguridad. La protección del código y software busca fortalecer las defensas contra posibles amenazas y garantizar que las aplicaciones sean resistentes a ataques malintencionados, contribuyendo a un entorno de desarrollo y adquisición de software seguro y confiable.

**1.1** **Prácticas de Desarrollo Seguro:**

La implementación de prácticas de desarrollo seguro se centra en inculcar medidas preventivas desde las etapas iniciales del ciclo de vida del software. Esto abarca desde la selección de lenguajes de programación seguros hasta la adopción de buenas prácticas de codificación, asegurando que el código resultante sea resistente a vulnerabilidades comunes.

**1.2** **Análisis Estático y Dinámico del Código:**

La realización de análisis estático y dinámico del código es esencial para identificar y mitigar posibles vulnerabilidades. El análisis estático evalúa el código sin ejecutarlo, mientras que el análisis dinámico evalúa el comportamiento del software en tiempo de ejecución, proporcionando una evaluación integral de la seguridad.

**1.3** **Integración de Medidas de Seguridad en el Ciclo de Vida:**

La integración de medidas de seguridad en cada fase del ciclo de vida del software asegura que la seguridad sea una consideración continua. Desde el diseño hasta la implementación y las actualizaciones, este enfoque garantiza que las decisiones de seguridad estén presentes en todo momento.

**1.4** **Pruebas de Seguridad:**

Las pruebas de seguridad, como las pruebas de penetración y otras evaluaciones, son cruciales para evaluar la resistencia del software frente a posibles amenazas. Estas pruebas permiten identificar y abordar posibles debilidades antes de la implementación en entornos de producción.

**1.5** **Protección contra Amenazas Comunes:**

La protección contra amenazas comunes, como inyecciones de código y otros ataques, implica la aplicación de contramedidas específicas para mitigar los riesgos. Esto incluye el uso de técnicas y controles específicos diseñados para prevenir vulnerabilidades conocidas.

**1.6** **Gestión de Identidades y Accesos Seguros:**

La gestión segura de identidades y accesos asegura que solo usuarios autorizados tengan acceso a recursos específicos. Esto incluye la implementación de controles de acceso adecuados y la gestión eficiente de credenciales para prevenir posibles abusos.

**1.7** **Evaluación de Diseños desde una Perspectiva de Seguridad:**

La evaluación de diseños desde una perspectiva de seguridad implica revisar las decisiones arquitectónicas y de diseño para identificar posibles riesgos. Este enfoque proactivo busca abordar vulnerabilidades potenciales antes de que se integren en el desarrollo completo.

# Modelo de Madurez

|  |  |
| --- | --- |
| NIVEL 1 [Impredecible y reactivo]: La actividad se completa, pero a menudo se retrasa y supera el presupuesto. | 1. Almacenar todo el código fuente y la configuración como código en un repositorio de código y restringir el acceso a él según la naturaleza del código. |
| NIVEL 2 [Gestionado a nivel de proceso]: Las actividades se planifican, ejecutan, miden y controlan. | 1. Utilizar las funciones de control de versiones del repositorio para realizar un seguimiento de todos los cambios realizados en el código con responsabilidad ante la cuenta individual. 2. Utilizar la firma de confirmación para repositorios de código. 3. Hacer que el propietario del código revise y apruebe todos los cambios realizados por otros en el código. 4. Utilizar la firma de código para ayudar a proteger la integridad de los ejecutables. 5. Validar todas las entradas y validar y codificar correctamente todas las salidas. 6. Evitar el uso de funciones y llamadas no seguras. 7. Detectar errores y manejarlos con elegancia. 8. buscar otras vulnerabilidades que sean comunes a los lenguajes y el entorno de desarrollo. |
| NIVEL 3 [Proactivo antes que reactivo]: Los estándares de toda la organización brindan orientación a través de proyectos, programas y carteras. | 1. Publicar hashes criptográficos para archivos de publicación en un sitio web bien seguro.  2. Revisar periódicamente los procesos de firma de código, incluida la renovación, rotación, revocación y protección de certificados.  3. Almacenar los archivos de lanzamiento, imágenes asociadas, etc., en repositorios siguiendo la política establecida por la organización. Permitir el acceso de solo lectura al personal necesario y ningún acceso a nadie más.  4. Almacenar y proteger la información de verificación de la integridad de la versión y los datos de procedencia, por ejemplo, manteniéndolos en una ubicación separada de los archivos de la versión o firmando los datos.  5. Proporcionar capacidades de registro y seguimiento.  6. Hacer que el desarrollador revise su propio código legible por humanos para complementar (no reemplazar) la revisión del código realizada por otras personas o herramientas. |
| NIVEL 4 [Gestionado Cuantitativamente]: La organización está basada en datos con objetivos cuantitativos de mejora del desempeño que son predecibles | 1. Poner los datos de procedencia a disposición de los adquirentes de software de acuerdo con las políticas de la organización, preferiblemente utilizando formatos basados en estándares. 2. Poner los datos de procedencia a disposición de las operaciones y los equipos de respuesta de la organización para ayudarlos a mitigar las vulnerabilidades del software. 3. Proteger la integridad de los datos de procedencia y proporcionar una forma para que los destinatarios verifiquen la integridad de los datos de procedencia. 4. Actualizar los datos de procedencia cada vez que se actualice alguno de los componentes del software. |
| NIVEL 5 [Optimizado]: La organización se centra en la mejora continua y está diseñada para girar y responder a las oportunidades y los cambios. |  |

1. **Preguntas de Auditoría para Protección del Código y Software**

|  |
| --- |
| Temática: Publicación de Hashes Criptográficos |
| Involucra una revisión exhaustiva de cómo se gestionan y divulgan los valores hash asociados al código fuente y binarios. Este proceso implica verificar la implementación correcta de algoritmos de hash seguros, como SHA-256, para garantizar la integridad del código durante la transmisión y almacenamiento. La auditoría se enfoca en confirmar que los hashes se generan de manera segura y que cualquier publicación o distribución se realiza mediante canales protegidos. Además, se evalúa la coherencia entre los hashes publicados y el contenido real del código, asegurando que no haya manipulaciones maliciosas. La gestión adecuada de los hashes criptográficos es esencial para verificar la autenticidad del software y prevenir la introducción de modificaciones no autorizadas que podrían comprometer la seguridad y la confianza en el código fuente.  Este levantamiento puede incluir análisis específicos tales como:   * ¿Cómo se generan los hashes criptográficos para el código y los binarios?   + ¿Se utilizan algoritmos de hash criptográficos fuertes y actualizados para generar los hashes?   + ¿El proceso de generación de hashes es resistente a ataques, como colisiones o preimágenes?   + ¿Se documenta y revisa regularmente la implementación técnica de la generación de hashes? * ¿Cómo se publican y divulgan los hashes criptográficos?   + ¿Los canales de publicación de los hashes, como sitios web o repositorios, utilizan conexiones seguras (por ejemplo, HTTPS)?   + ¿Existen medidas para garantizar la integridad de los hashes durante la publicación y transmisión?   + ¿Se auditan regularmente los procesos de publicación para detectar posibles manipulaciones? * ¿Cómo se protege la generación y gestión de claves criptográficas asociadas a los hashes?   + ¿Las claves criptográficas se almacenan y gestionan en un entorno seguro, como un HSM (Módulo de Seguridad de Hardware)?   + ¿Se implementan controles de acceso estrictos para la generación y el manejo de claves criptográficas?   + ¿Existe un proceso documentado para la rotación y revocación de claves criptográficas? * ¿Cómo se garantiza la coherencia entre los hashes publicados y el contenido real del código o binario?   + ¿Se realizan verificaciones automáticas para comparar los hashes publicados con los obtenidos directamente del código o binario?   + ¿Hay un proceso establecido para reportar y corregir cualquier discrepancia encontrada entre los hashes publicados y el contenido real?   + ¿Se documentan los procedimientos de verificación y se revisan periódicamente? * ¿Cómo se actualizan los hashes cuando hay cambios en el código o binarios?   + ¿El proceso de actualización de hashes es automático y se activa inmediatamente después de cualquier cambio en el código o binarios?   + ¿Se asegura que los nuevos hashes se publiquen de manera coherente con las actualizaciones del software?   + ¿Existen controles para evitar la publicación de hashes incorrectos o desactualizados? * ¿Cómo se comunica la importancia de verificar hashes a los usuarios?   + ¿Se proporcionan instrucciones claras en la documentación para que los usuarios sepan cómo verificar los hashes?   + ¿Se promueve activamente la verificación de hashes como una práctica estándar en comunicaciones con los usuarios?   + ¿Se realizan campañas de sensibilización para educar a los usuarios sobre los riesgos de no verificar los hashes? |

|  |
| --- |
| Temática: Proporción de Capacidades de Registro y Seguimiento |
| Implica llevar a cabo una evaluación exhaustiva de cómo se implementan y utilizan los registros y sistemas de seguimiento dentro del entorno de desarrollo y adquisición de software. Este proceso abarca la revisión de la amplitud y profundidad de los registros generados durante el desarrollo, compilación, despliegue y ejecución de aplicaciones. La auditoría se enfoca en confirmar que los registros sean suficientemente detallados para identificar posibles incidentes de seguridad, errores de programación y actividades anómalas. Además, se evalúa la capacidad de correlacionar y analizar los registros para detectar patrones y comportamientos inusuales. La proporción adecuada de capacidades de registro y seguimiento es esencial para facilitar la respuesta rápida ante incidentes, la detección temprana de amenazas y la mejora continua de la postura de seguridad en el ciclo de vida del software.  Este levantamiento puede incluir análisis específicos tales como:   * ¿Cómo se definen y documentan las capacidades de registro en el proceso de desarrollo de software?   + ¿Existen requisitos claros y bien documentados que especifican qué eventos deben registrarse durante el desarrollo del software?   + ¿Están los requisitos de registro alineados con las mejores prácticas de seguridad y normativas de cumplimiento aplicables?   + ¿Se revisan y actualizan regularmente estos requisitos y la documentación relacionada? * ¿Cómo se implementan los registros durante el ciclo de vida del software, desde el desarrollo hasta la producción?   + ¿Los procesos de desarrollo incluyen la generación automática y consistente de registros en todas las fases?   + ¿Se verifica la integridad y exhaustividad de los registros durante el despliegue y la ejecución del software en producción?   + ¿Existen mecanismos para asegurar que los registros no se pierdan durante la transición entre fases del ciclo de vida del software? * ¿Cómo se asegura la calidad y relevancia de la información registrada en los logs?   + ¿Los registros incluyen información suficiente y detallada para la detección de amenazas y la investigación de incidentes?   + ¿Se analizan periódicamente muestras de registros para verificar su consistencia y utilidad?   + ¿Se implementan controles para asegurar que la información registrada sea relevante y no redundante? * ¿Cómo se gestiona la privacidad y la sensibilidad de los datos registrados?   + ¿Se aplican políticas y controles específicos para anonimizar o proteger la información sensible en los registros?   + ¿Los procesos de registro cumplen con las regulaciones de privacidad aplicables, como GDPR o CCPA?   + ¿Existe una revisión regular de los registros para asegurar que no se estén registrando datos sensibles innecesarios? * ¿Cómo se almacenan y protegen los registros para prevenir alteraciones no autorizadas?   + ¿Se utilizan controles de acceso y medidas de seguridad robustas para proteger los sistemas de almacenamiento de registros?   + ¿Se implementan mecanismos de auditoría para detectar y prevenir alteraciones no autorizadas en los registros?   + ¿Los registros se almacenan en un entorno seguro que garantiza su integridad y confidencialidad? * ¿Cómo se correlacionan y analizan los registros para detectar patrones y comportamientos inusuales?   + ¿Se utilizan herramientas y procesos de correlación para analizar los eventos registrados y detectar posibles amenazas?   + ¿Las herramientas de análisis permiten identificar indicadores de compromiso y comportamientos anómalos?   + ¿Existen procesos documentados para la revisión y análisis continuo de los registros en busca de patrones inusuales? * ¿Cómo se garantiza la disponibilidad y accesibilidad de los registros cuando se necesitan para investigar incidentes?   + ¿Se han implementado procedimientos de respaldo y redundancia para asegurar la disponibilidad de los registros?   + ¿Los registros son accesibles para los equipos de seguridad e investigación en caso de un incidente?   + ¿Se revisan regularmente los procedimientos de acceso para asegurar que los registros estén disponibles durante y después de un incidente? |

|  |
| --- |
| Temática: Protección de la integridad de los datos de procedencia |
| Implica una revisión exhaustiva de cómo se asegura la integridad y autenticidad de los datos relacionados con la procedencia del código fuente y los binarios. Este proceso abarca la evaluación de las medidas implementadas para prevenir la manipulación maliciosa de información crítica, como los registros de cambios, autorías y versiones del código. Se examina la implementación de firmas digitales, control de acceso y sistemas de gestión de versiones para garantizar que los datos de procedencia no sean alterados de manera no autorizada. La auditoría se centra en confirmar que la trazabilidad y la integridad de estos datos sean preservadas, contribuyendo a la confiabilidad del software y a la capacidad de rastrear cualquier cambio o modificación en el código a lo largo del tiempo.  Este levantamiento puede incluir análisis específicos tales como:   * ¿Cómo se gestiona el control de acceso a los datos de procedencia, como registros de cambios y versiones de código fuente?   + ¿Se han definido y aplicado políticas claras de control de acceso que limitan el acceso a los datos de procedencia solo a personal autorizado?   + ¿Los permisos de acceso se revisan y actualizan regularmente para reflejar cambios en los roles o responsabilidades?   + ¿Se mantienen registros de auditoría detallados para rastrear quién accede o modifica los datos de procedencia? * ¿Cómo se implementan las firmas digitales o hash para verificar la autenticidad e integridad de los datos de procedencia?   + ¿Las firmas digitales o los hashes se aplican consistentemente a todos los datos de procedencia relevantes?   + ¿Existen procedimientos para la verificación regular de la integridad de los datos utilizando estas firmas o hashes?   + ¿Se utilizan algoritmos criptográficos fuertes y actualizados para generar las firmas digitales o hashes? * ¿Cómo se gestionan y auditan las autorías y cambios en el código fuente a lo largo del tiempo?   + ¿Se registran todas las modificaciones en el código fuente, incluyendo la autoría y la razón del cambio?   + ¿Se realizan auditorías periódicas para asegurar que todos los cambios se registren y autoricen correctamente?   + ¿El historial de versiones del código fuente es completo y accesible para los equipos relevantes? * ¿Cómo se protegen los sistemas de gestión de versiones (VCS) para prevenir accesos no autorizados o manipulación de datos?   + ¿Los VCS están configurados con políticas de acceso estrictas que limitan quién puede acceder o modificar los datos de procedencia?   + ¿Se utilizan medidas de seguridad, como autenticación multifactor y cifrado, para proteger los VCS?   + ¿Se revisan y actualizan regularmente las configuraciones de seguridad y políticas de acceso en los VCS? * ¿Cómo se asegura la integridad de los datos de procedencia durante su transmisión o distribución?   + ¿Se utilizan protocolos de seguridad, como TLS, para asegurar la transmisión de los datos de procedencia?   + ¿Se implementan métodos de cifrado para proteger los datos durante su distribución?   + ¿Existen mecanismos para verificar que los datos no hayan sido alterados durante la transmisión? * ¿Cómo se educan y sensibilizan a los desarrolladores sobre la importancia de la integridad de los datos de procedencia?   + ¿Se proporcionan programas de formación regulares que incluyan temas sobre la importancia de proteger la integridad de los datos de procedencia?   + ¿Existen materiales educativos y sesiones informativas accesibles para todos los desarrolladores?   + ¿Se promueve activamente una cultura de seguridad entre los desarrolladores respecto a la integridad de los datos? * ¿Cómo se realizan auditorías periódicas para verificar la integridad y autenticidad de los datos de procedencia? * ¿Se realizan auditorías programadas para revisar la integridad y autenticidad de los datos de procedencia?   + ¿Los informes de auditoría se revisan y se actúa sobre cualquier anomalía detectada? * ¿Se realizan pruebas regulares para asegurar que los controles implementados sigan siendo efectivos? |